**Projet 1 :** Création d’un Dashboard pour la visualisation en temps réel des données météorologiques en utilisant Kafka, Spark, Hive, Tableau et Apache Airflow.

**Table des matières**

[I. Objectif du projet 1](#_Toc153039606)

[II. Prérequis pour le projet - Installation et Configuration 1](#_Toc153039607)

[1. Environnement de Développement 1](#_Toc153039608)

[2. Configuration de l'Environnement Dockerisé 2](#_Toc153039609)

[a. Création des Conteneurs 2](#_Toc153039610)

[b. Exécution des conteneurs 5](#_Toc153039611)

[c. Exécution d’un conteneur spécifique 6](#_Toc153039612)

[3. Installation et Configuration de Tableau 6](#_Toc153039613)

[a. Téléchargement de Tableau Desktop : 6](#_Toc153039614)

[b. Activation de Tableau : 7](#_Toc153039615)

[c. Connexion à Hive depuis Tableau : 7](#_Toc153039616)

[d. Sélectionner la Table ou la Vue Hive : 9](#_Toc153039617)

[e. Configuration des Connexions de Données : 9](#_Toc153039618)

[III. Création du Dashboard pour la visualisation 10](#_Toc153039619)

[1. Obtention de la clé d'API OpenWeatherMap 10](#_Toc153039620)

[2. Collecte des Données Météorologiques depuis l'api vers un topic Kafka 11](#_Toc153039621)

[3. Traitement et Stockage des Données avec Spark et Hive 13](#_Toc153039622)

[4. Visualisation des Résultats avec Tableau 15](#_Toc153039623)

[a. Importation des Données 15](#_Toc153039624)

[b. Création des Visualisations 16](#_Toc153039625)

[c. Création de Tableaux de Bord (Dashboard) 18](#_Toc153039626)

[5. Orchestration avec Apache Airflow 18](#_Toc153039627)

[a. Interface utilisateur 18](#_Toc153039628)

[b. Création et exécution du dag 19](#_Toc153039629)

# I. Objectif du projet

Ce projet vise à mettre en place un système intégré pour le traitement et la visualisation en temps réel des données météorologiques en utilisant diverses technologies. Il débute par la collecte de données depuis l'API OpenWeatherMap. Ces données transitent ensuite par Kafka pour un traitement en temps réel via Spark Streaming. Les résultats obtenus sont stockés dans Hive, une base de données d'entrepôt dédiée. Enfin, ces données sont exploitées visuellement grâce à Tableau. L'intégralité de ce processus est déployée dans un environnement Dockerisé, simplifiant la gestion et le déploiement de l'infrastructure essentielle à cette chaîne de traitement des données météorologiques.

# II. Prérequis pour le projet - Installation et Configuration

Ce projet nécessite l'installation et la configuration préalable de plusieurs outils et environnements. Suivez attentivement ces étapes pour garantir un déroulement sans accroc du projet.

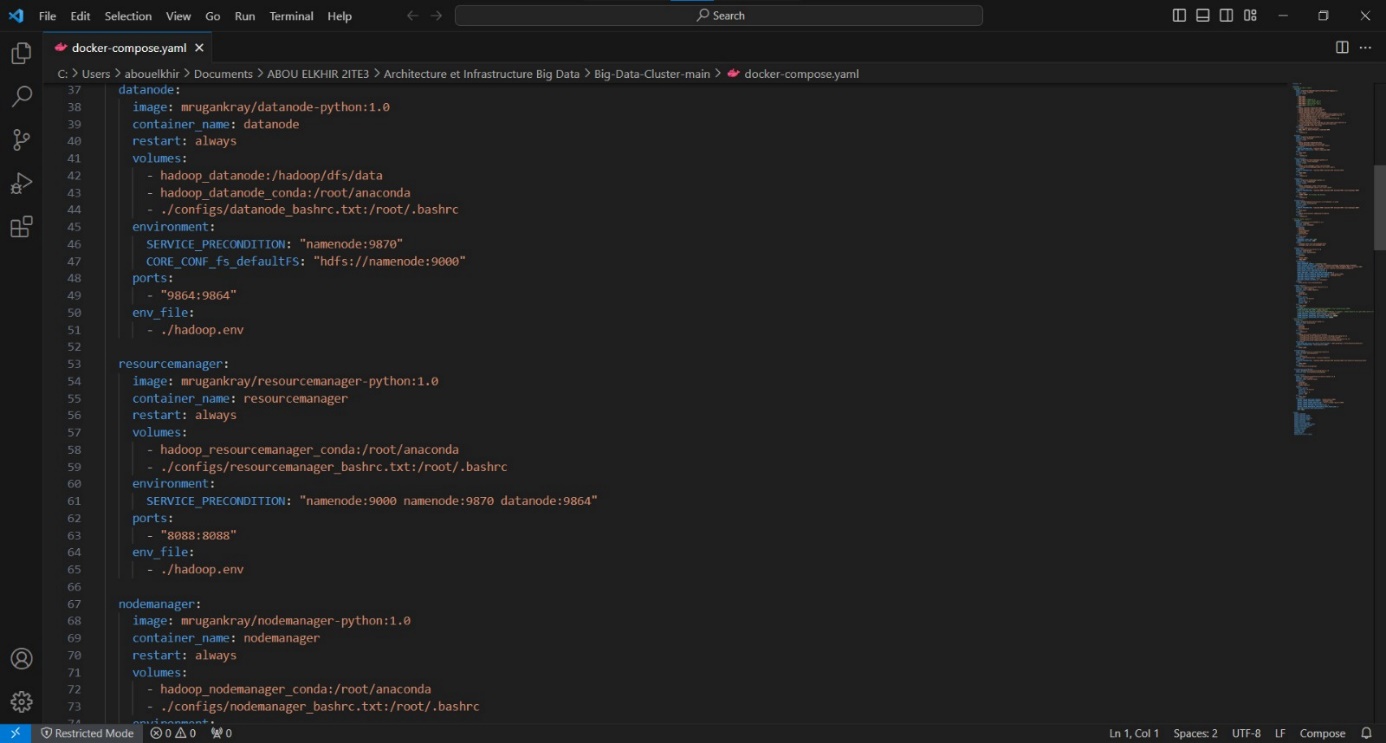
## 1. Environnement de Développement

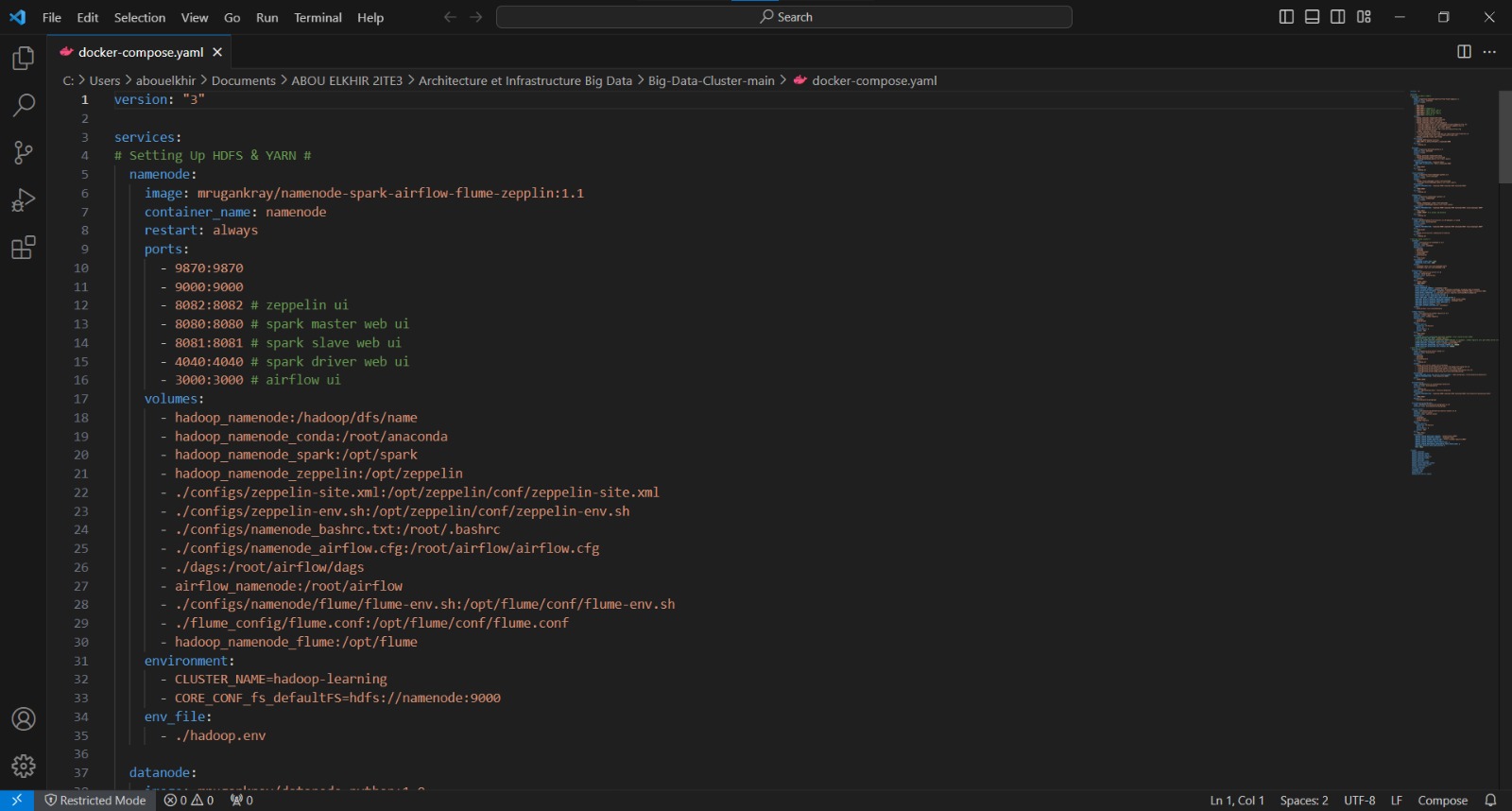
1. Système d'Exploitation : Vérifiez que vous disposez d'un système d'exploitation compatible avec les outils nécessaires (Exemple : Ici, on utilise Windows).
2. RAM : Assurez-vous d'avoir une mémoire RAM supérieure à 13 Go.
3. Docker :
   * Téléchargez et installez Docker en suivant les instructions spécifiques à votre système d'exploitation : [lien vers Docker](https://docs.docker.com/get-docker/)
   * Vérifiez l'installation avec la commande : docker –version
4. Docker Compose :
   * Téléchargez et installez Docker Compose en suivant les instructions spécifiques à votre système d'exploitation : [lien vers Docker Compose](https://docs.docker.com/compose/install/)
   * Vérifiez l'installation avec la commande : docker-compose --version

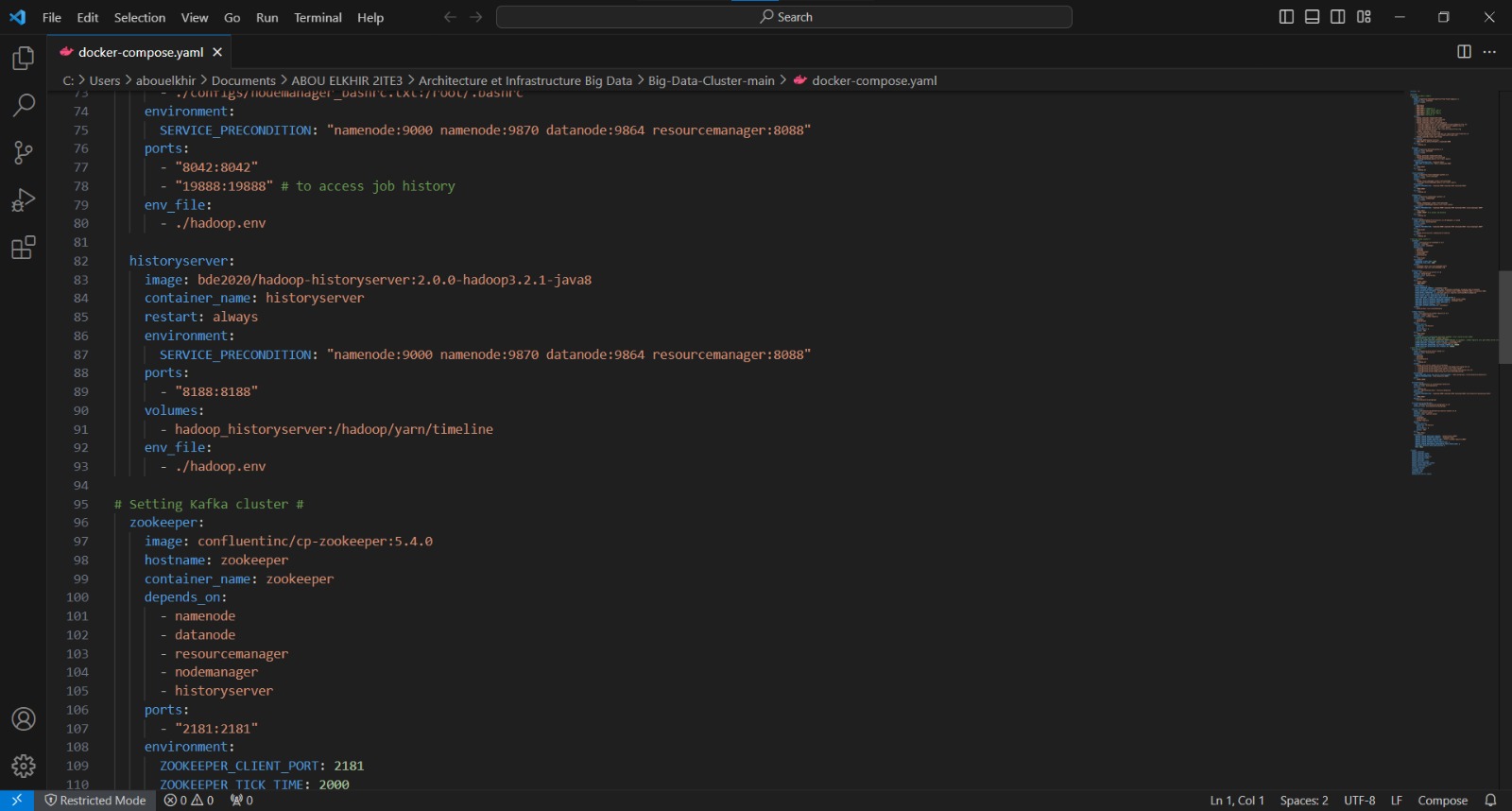
## 2. Configuration de l'Environnement Dockerisé

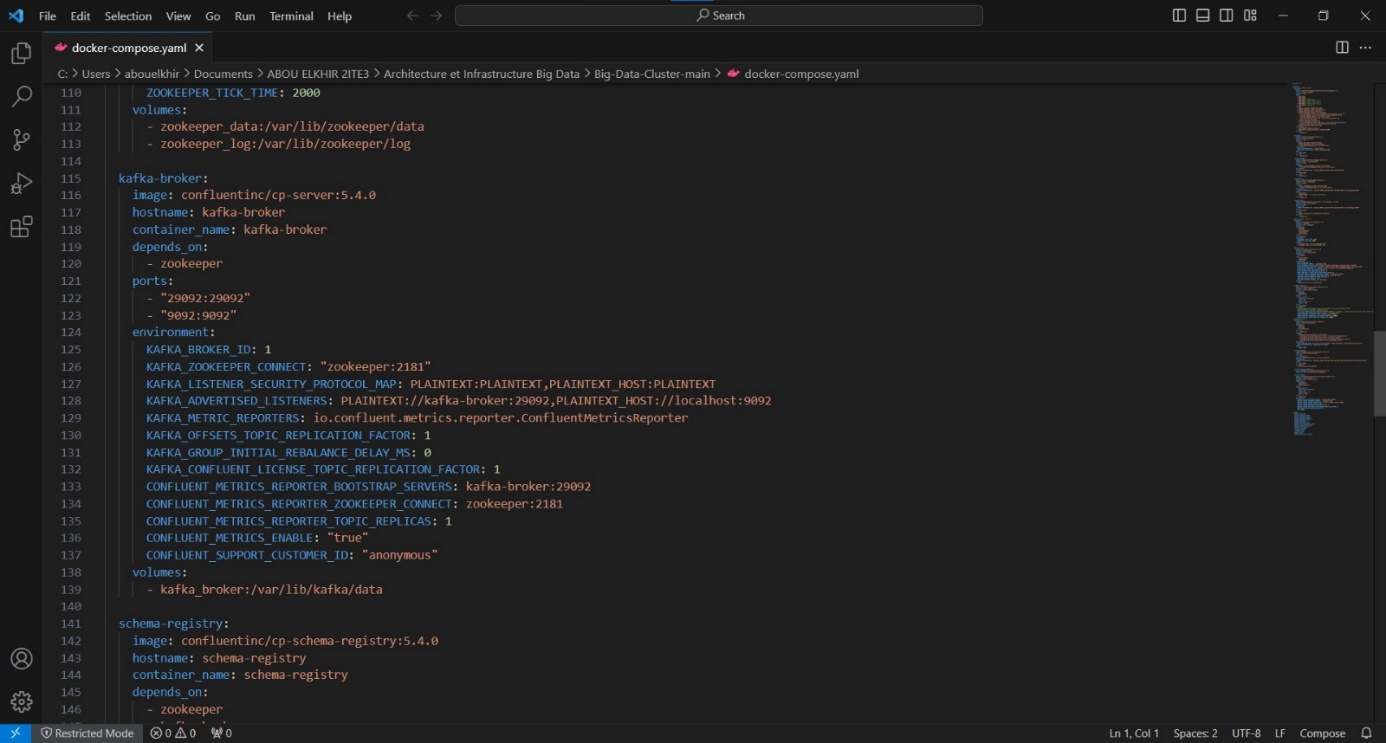
### Création des **Conteneurs**

* + Élaborez un fichier docker-compose.yml décrivant les services nécessaires pour cette application.Se referer au fichier docker-compose.yml dans le dossier du projet.









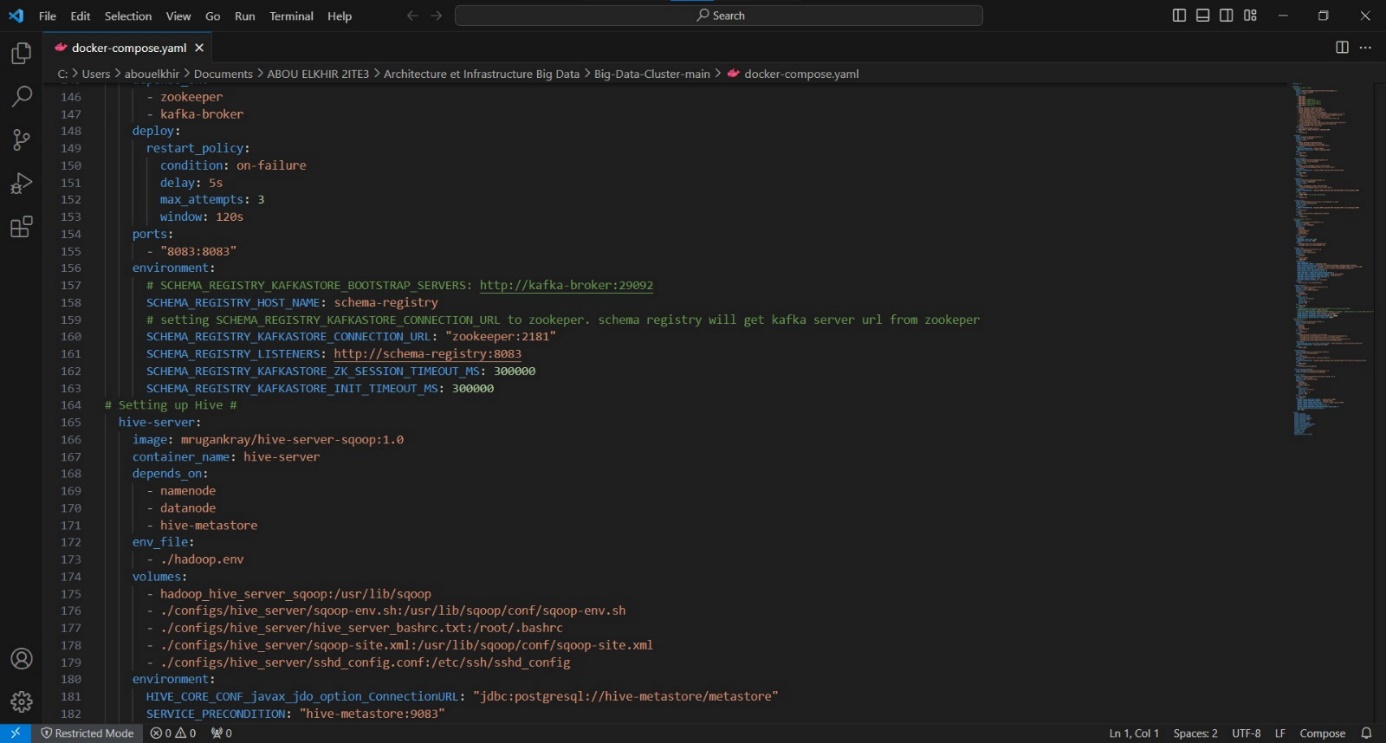






Figure 1.1 Code du fichier docker-compose pour la création des conteneurs docker

### Exécution des conteneurs

Pour démarrer les conteneurs définis dans votre fichier **docker-compose.yml** et les exécuter en arrière-plan, utilisez la commande suivante : docker-compose up -d

Une fois les conteneurs lancés, vous pouvez lister les identifiants des conteneurs en cours d’exécution et leurs ports associés en utilisant la commande : docker ps

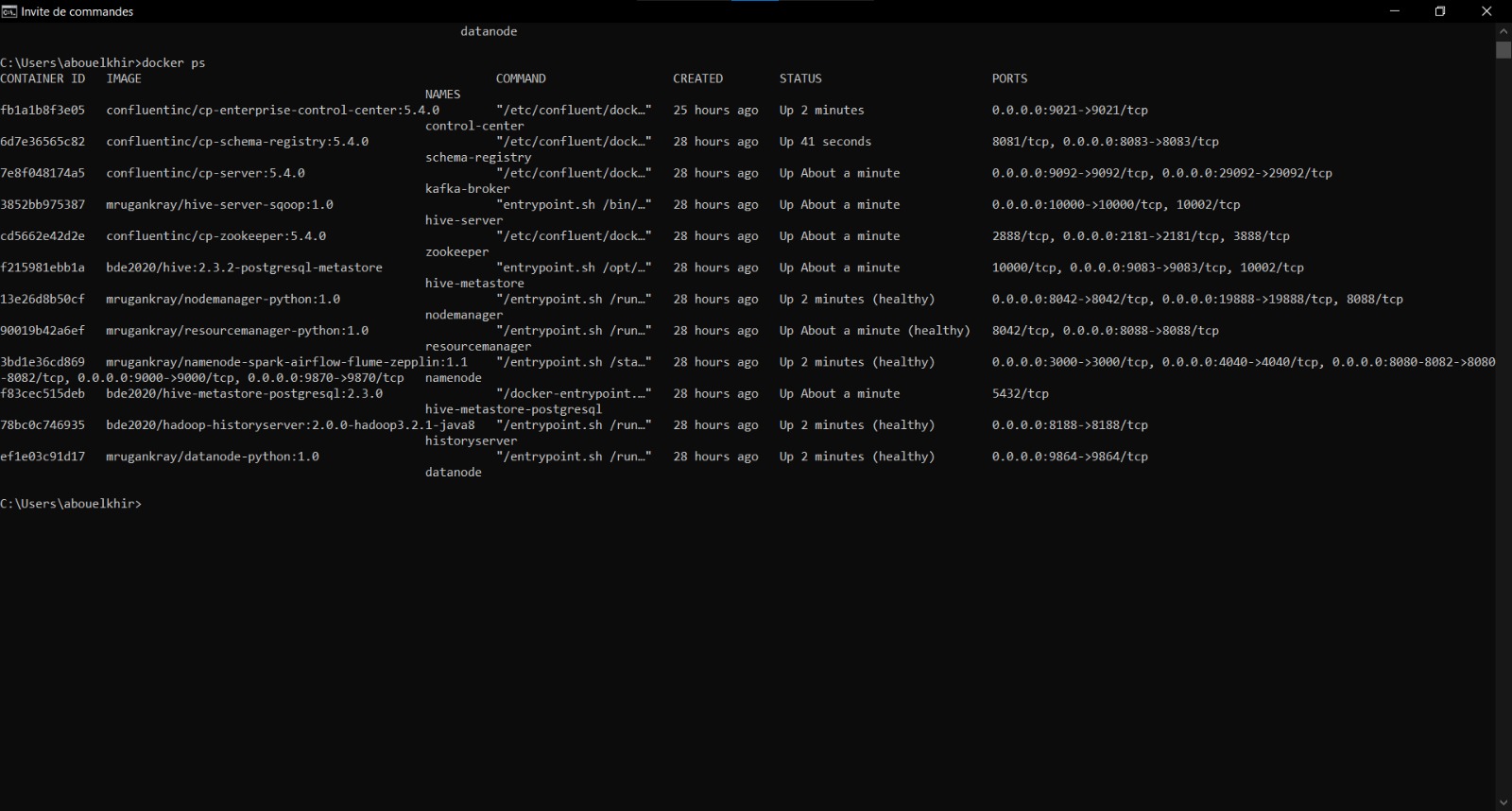


Figure 1.2 Affichage de la liste des conteneurs

Pour accéder aux services dans un navigateur web, utilisez les URL suivantes :

* Zeppelin : [http://localhost:8080](http://localhost:8080/)
* Kafka Broker : [http://localhost:29092](http://localhost:29092/)
* Spark : [http://localhost:8080](http://localhost:8080/)
* Airflow : [http://localhost:3000](http://localhost:3000/)
* Namenode : [http://localhost:5432](http://localhost:5432/)

Assurez-vous que les services sont correctement démarrés et que les ports spécifiés dans votre fichier **docker-compose.yml** ne sont pas utilisés par d'autres applications sur votre système.

### Exécution d’un conteneur spécifique

Pour accéder au shell d'un conteneur spécifique en mode bash, utilisez la commande suivante, en remplaçant **id\_container** par l'identifiant du conteneur souhaité :

docker exec -it id\_container /bin/bash



Figure 1.3 Exécution d'un conteneur spécifique

## 3. Installation et Configuration de Tableau

Dans le cadre de ce projet, nous utilisons Tableau Desktop. Suivez ces étapes pour installer et configurer Tableau Desktop sur votre machine.

### Téléchargement de Tableau Desktop :

* Rendez-vous sur le site officiel de Tableau : [Tableau Download](https://www.tableau.com/products/desktop/download)

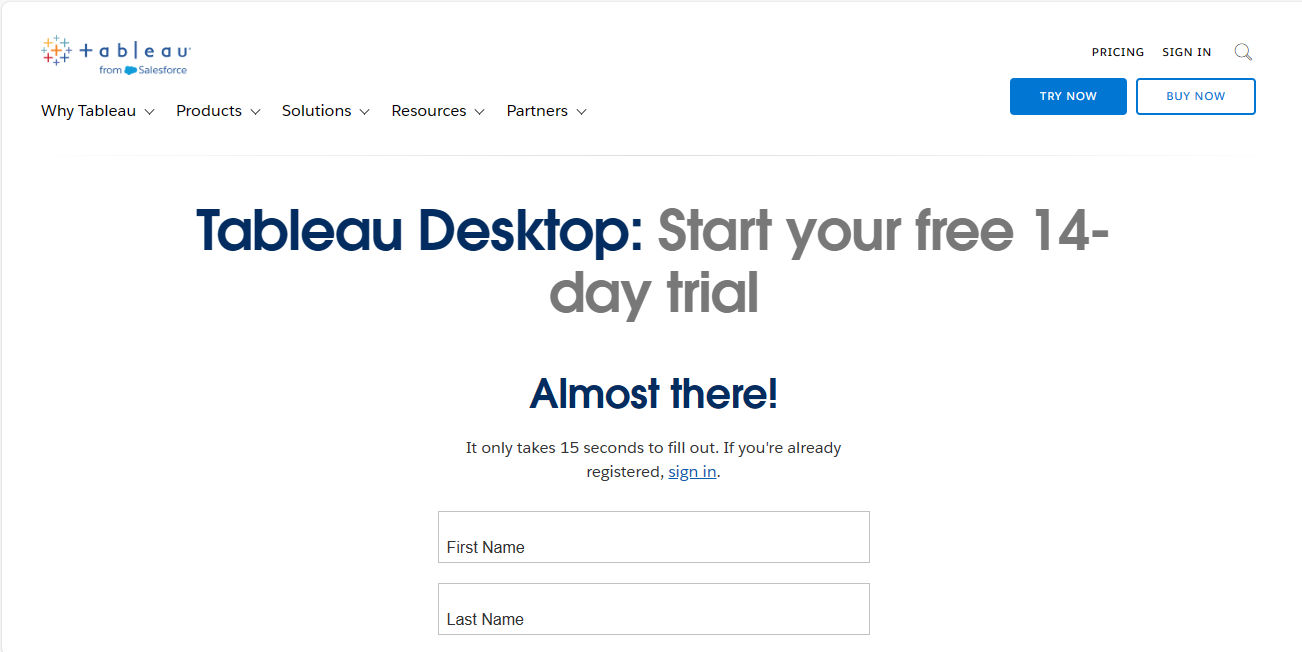


Figure 1.4 Site de téléchargement de tableau desktop

* Téléchargez la version appropriée de Tableau Desktop pour votre système d'exploitation (Windows ou Mac).
* Installez Tableau en suivant les instructions fournies lors du processus d'installation.

### Activation de Tableau :

* Lors du premier lancement, Tableau Desktop vous demandera d'activer le produit. Entrez votre clé de licence si vous en avez une, ou choisissez la version d'essai.

### Connexion à Hive depuis Tableau :

Pour connecter Tableau à Hive, vous pouvez utiliser le pilote ODBC (Open Database Connectivity) pour Hive. Suivez ces étapes générales pour effectuer cette connexion.

* Téléchargez et installez un pilote ODBC compatible avec Hive sur votre machine. Un exemple courant est le pilote Apache Hive ODBC Driver, que vous pouvez trouver sur le site de CDATA : [CDATA Hive ODBC Driver](https://www.cdata.com/drivers/hive/odbc/).

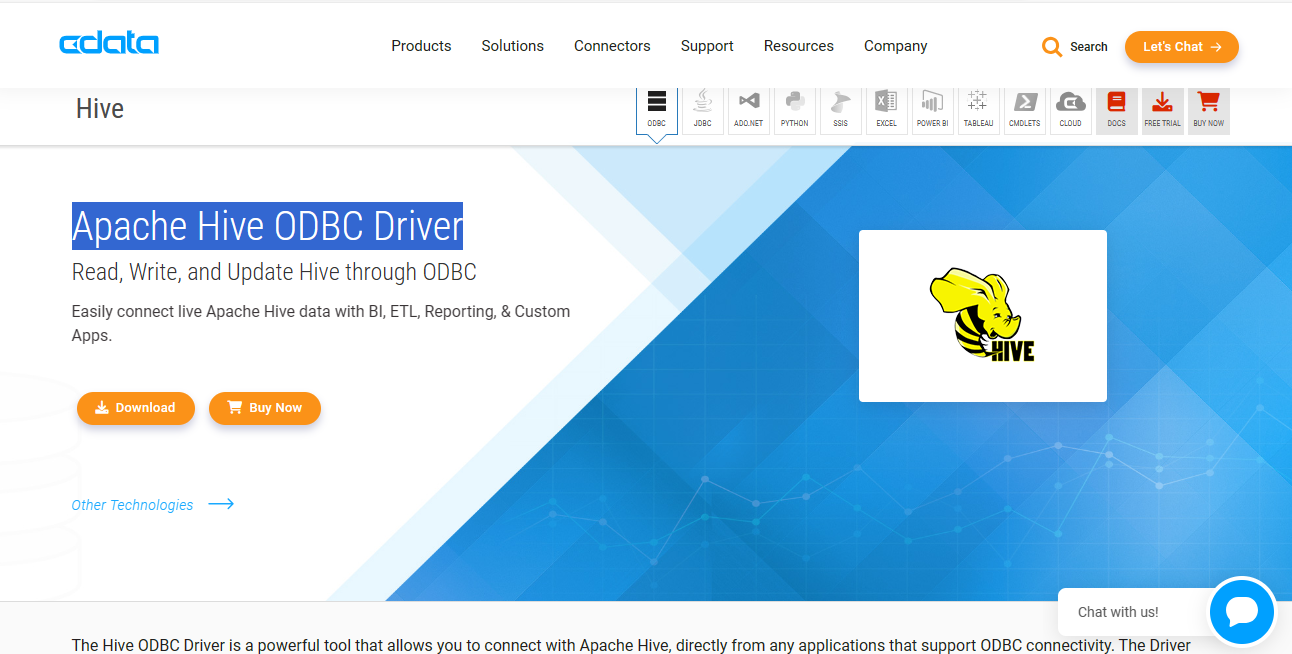


Figure 1.5 Site de téléchargement de ODBC pour Hive

* Après l'installation du pilote ODBC, configurez-le en fournissant les détails de connexion à votre cluster Hive, tels que l'adresse du serveur Hive, le port, le nom d'utilisateur, le mot de passe, etc.
* Dans Tableau Desktop, sous l'onglet "Données", choisissez "ODBC" comme type de connexion.
* Sélectionnez le pilote ODBC pour Hive que vous avez installé.
* Entrez les informations de connexion nécessaires, telles que le nom du serveur Hive, le port, le nom d'utilisateur et le mot de passe.

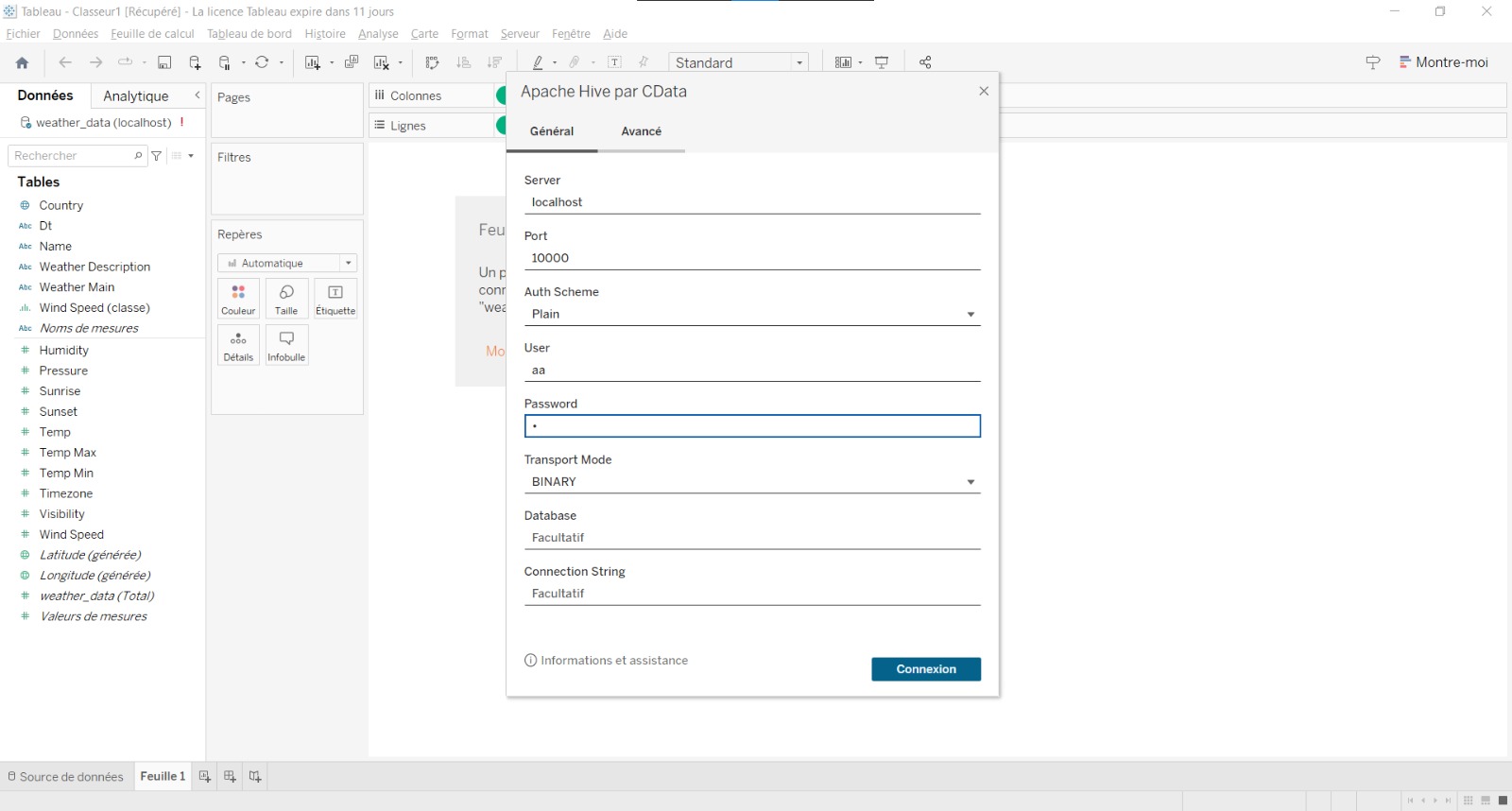


Figure 1.6 Etablissement de la connexion entre Hive et Tableau

* En fonction du pilote que vous utilisez, il peut y avoir des options supplémentaires à configurer. Consultez la documentation du pilote pour plus de détails.
* Après avoir configuré les paramètres de connexion, cliquez sur le bouton "Connecter" pour établir la connexion entre Tableau et Hive.

**Note :** Par défaut, aucun utilisateur n’est configuré sur Hive. Vous devez définir ici un utilisateur et un mot de passe que vous utiliserez à chaque fois que vous établirez la connexion.

### Sélectionner la Table ou la Vue Hive :

Une fois connecté, vous pourrez voir les bases de données Hive disponibles. Sélectionnez la base de données, puis choisissez la table ou la vue Hive que vous souhaitez analyser dans Tableau. Pour le moment vous il n’y a aucune table dans la base de données par défaut de hive.

### Configuration des Connexions de Données :

Vous pouvez configurer les connexions de données pour qu'elles soient dynamiques et mises à jour automatiquement en fonction des changements dans vos sources de données.

# III. Création du Dashboard pour la visualisation

## Obtention de la clé d'API OpenWeatherMap

* Accédez au site web d’OpenWeatherMap: [OpenWeatherMap](https://openweathermap.org/api).

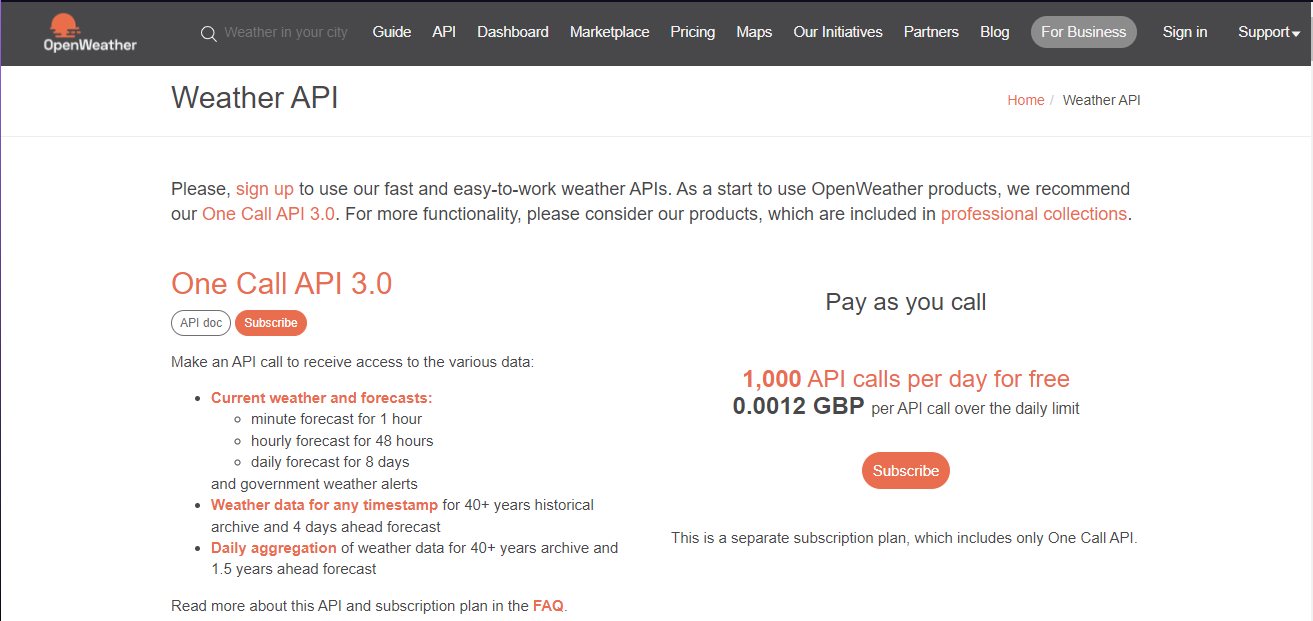


Figure 3.1 Site de OpenWeather

* Cliquez sur "Sign Up" (S'inscrire) pour créer un nouveau compte.
* Connectez-vous à votre compte OpenWeatherMap en utilisant les identifiants que vous avez créés.
* Une fois connecté, recherchez la section "My API keys" (clés API) dans votre espace utilisateur.

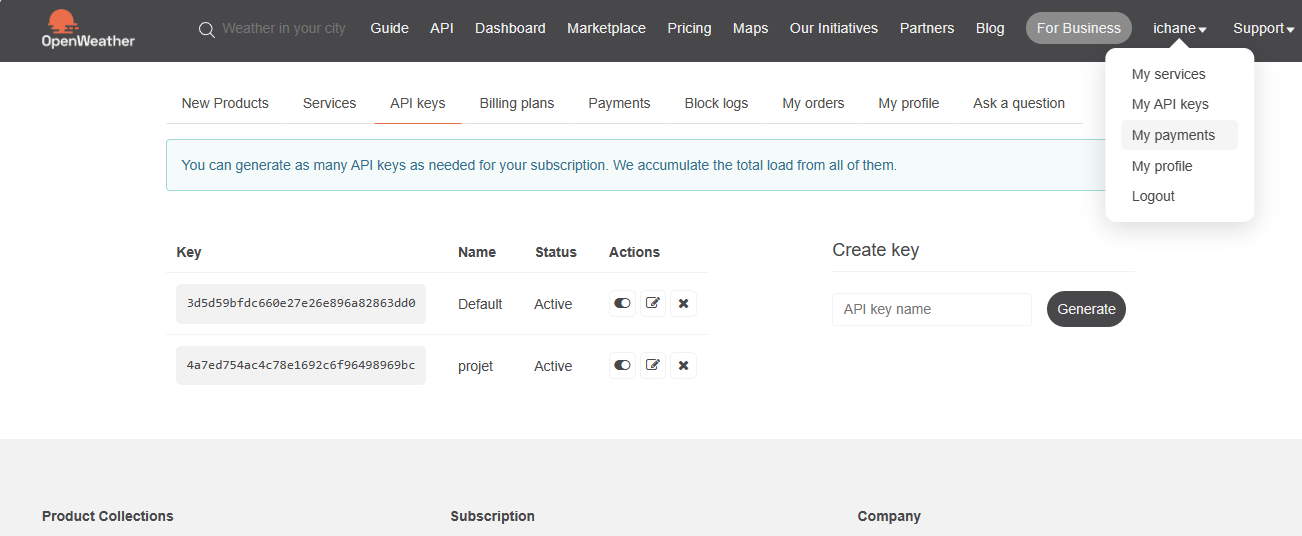


Figure 3.2 Obtention de la clé d'api

* Une fois la clé générée, copiez-la. Cette clé sera utilisée dans vos requêtes API pour identifier votre compte et autoriser l'accès aux données météorologiques.

## Collecte des Données Météorologiques depuis l'api vers un topic Kafka

Pour collecter des données météorologiques depuis l'API OpenWeatherMap et les publier sur un topic Kafka en utilisant Python, vous pouvez utiliser la bibliothèque requests pour effectuer la requête API et la bibliothèque kafka-python pour publier les données sur Kafka. Assurez-vous d'installer ces bibliothèques en utilisant la commande suivante si elles ne sont pas déjà installées.

* Créer un notebook python sur zeppelin et coller le code de collecte des données depuis l’api. Vous trouverez le code dans le fichier kafka\_stream dans le dossier du projet.
* Assurer vous de remplacer la clé d’api par la vôtre.
* Assurez-vous que Kafka est en cours d'exécution et accessible depuis Zeppelin.

Note : en fonction de votre api : gratuit ou payant vous pouvez avoir accès à différents types de données. Assurer vous donc d’adapter le code en fonction de la structure des données météorologiques que vous souhaitez collecter.

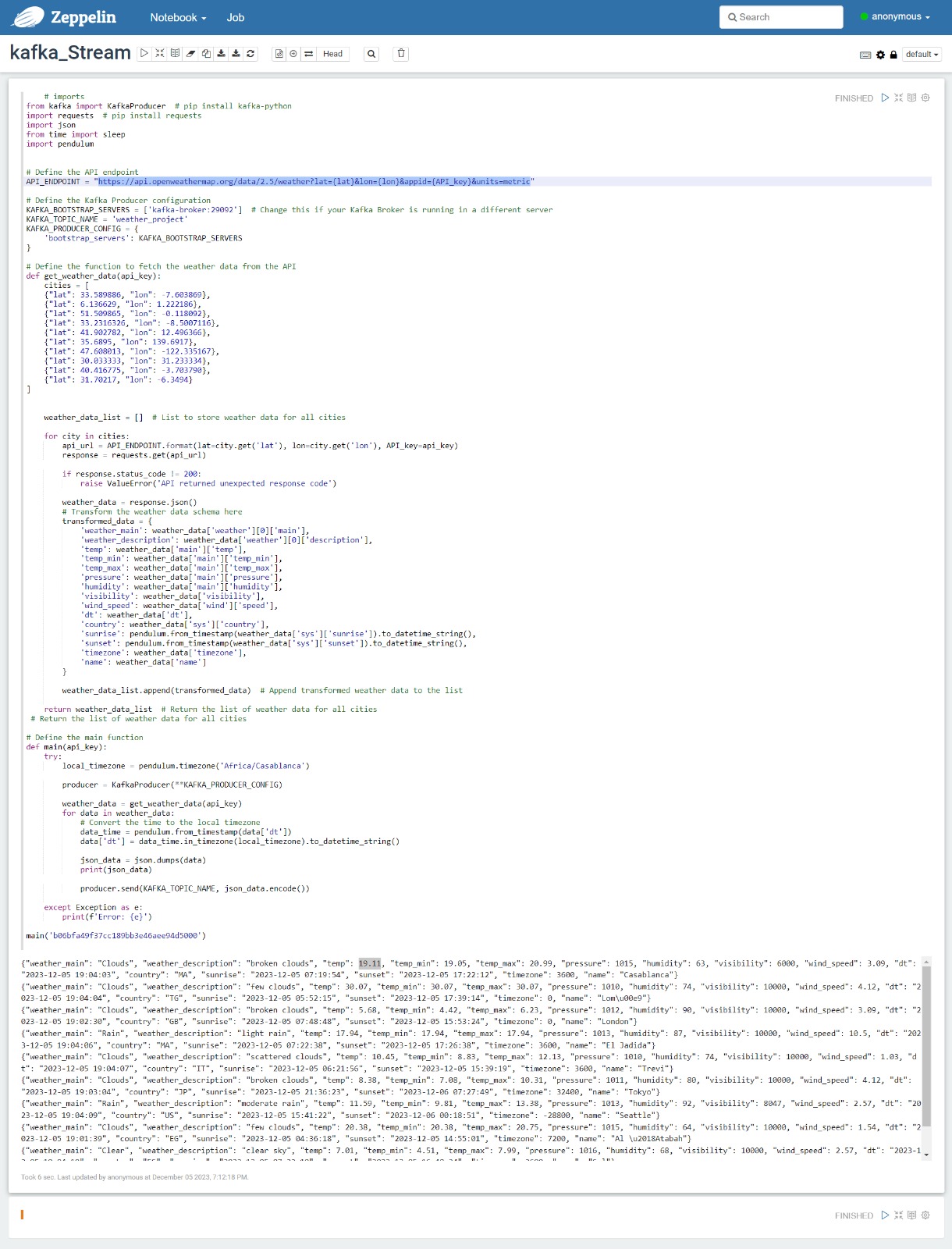


Figure 3.3 Code de collecte des données depuis l'api vers un topic Kafka

Les données /les messages peuvent etre visualiser au niveau du control center à l’adresse ([**http://localhost:9021**](http://localhost:9021))

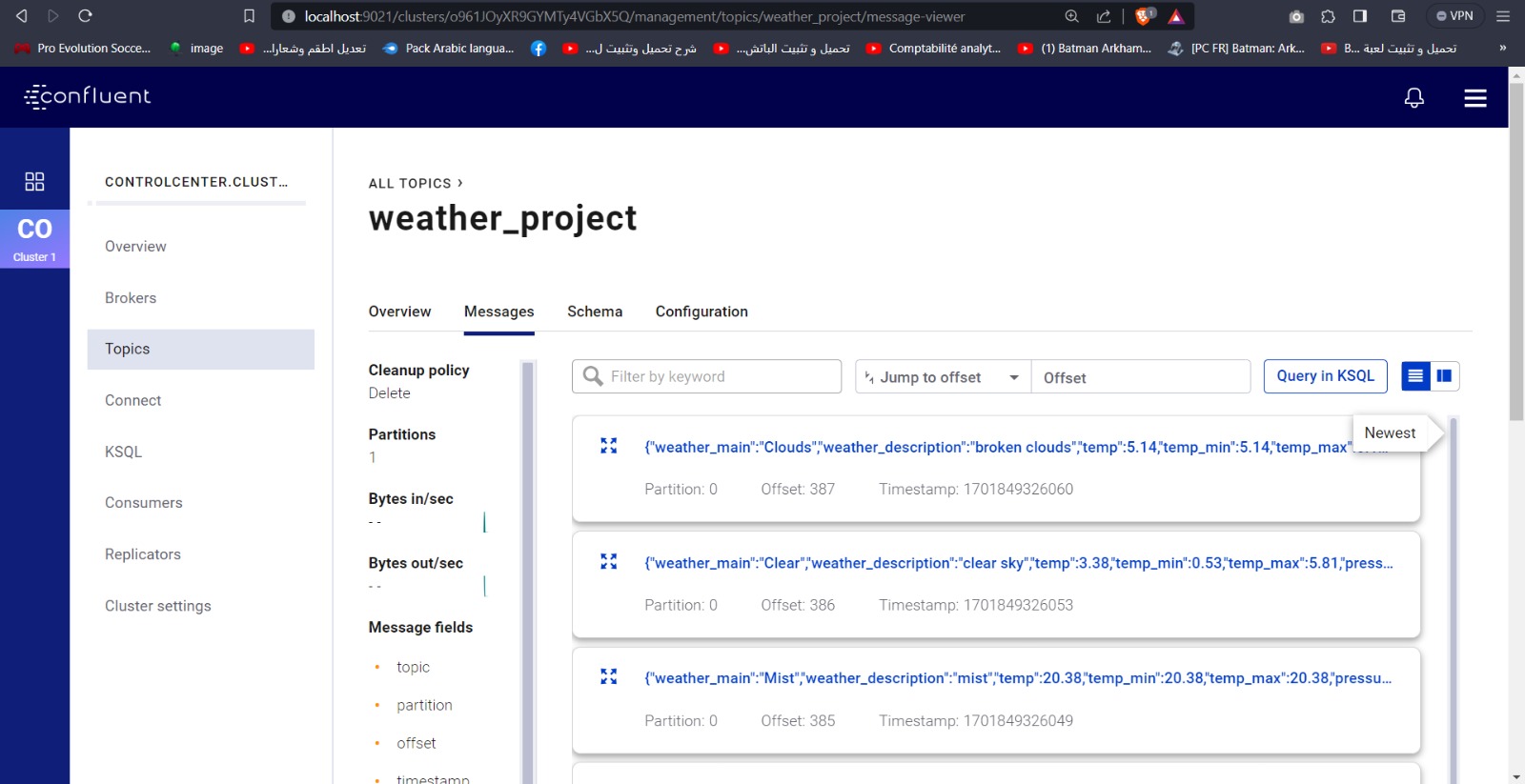


Figure 3.4 Visualisation des données du topic Kafka sur le control center

## Traitement et Stockage des Données avec Spark et Hive

Pour effectuer le traitement et le stockage des données avec Spark et Hive en utilisant PySpark dans Zeppelin, vous pouvez suivre ces étapes.

Assurez-vous que votre environnement est correctement configuré pour utiliser PySpark avec Zeppelin et que tout fonctionne correctement.

Le code à la *Figure 3.5* orchestre la réception de données depuis le topic Kafka, lesquelles sont soumises à des opérations de traitement en temps réel à l'aide de Spark Streaming. Une fois ces traitements effectués, les résultats obtenus sont consignés et enregistrés dans Hive, permettant ainsi une structuration et une conservation optimales de ces données traitées au sein de la base de données Hive.

Créer un nouveau notebook PySpark sur zeppelin et coller le code à exécuter. Vous trouverez le code dans le ficher spark\_streaming dans le dossier du projet.



Figure 3.5 Code pour récupérer les données du topic et les envoyer vers Hive

Pour vérifier que les données ont été correctement stockées dans Hive, accéder à l’interface de Hadoop.

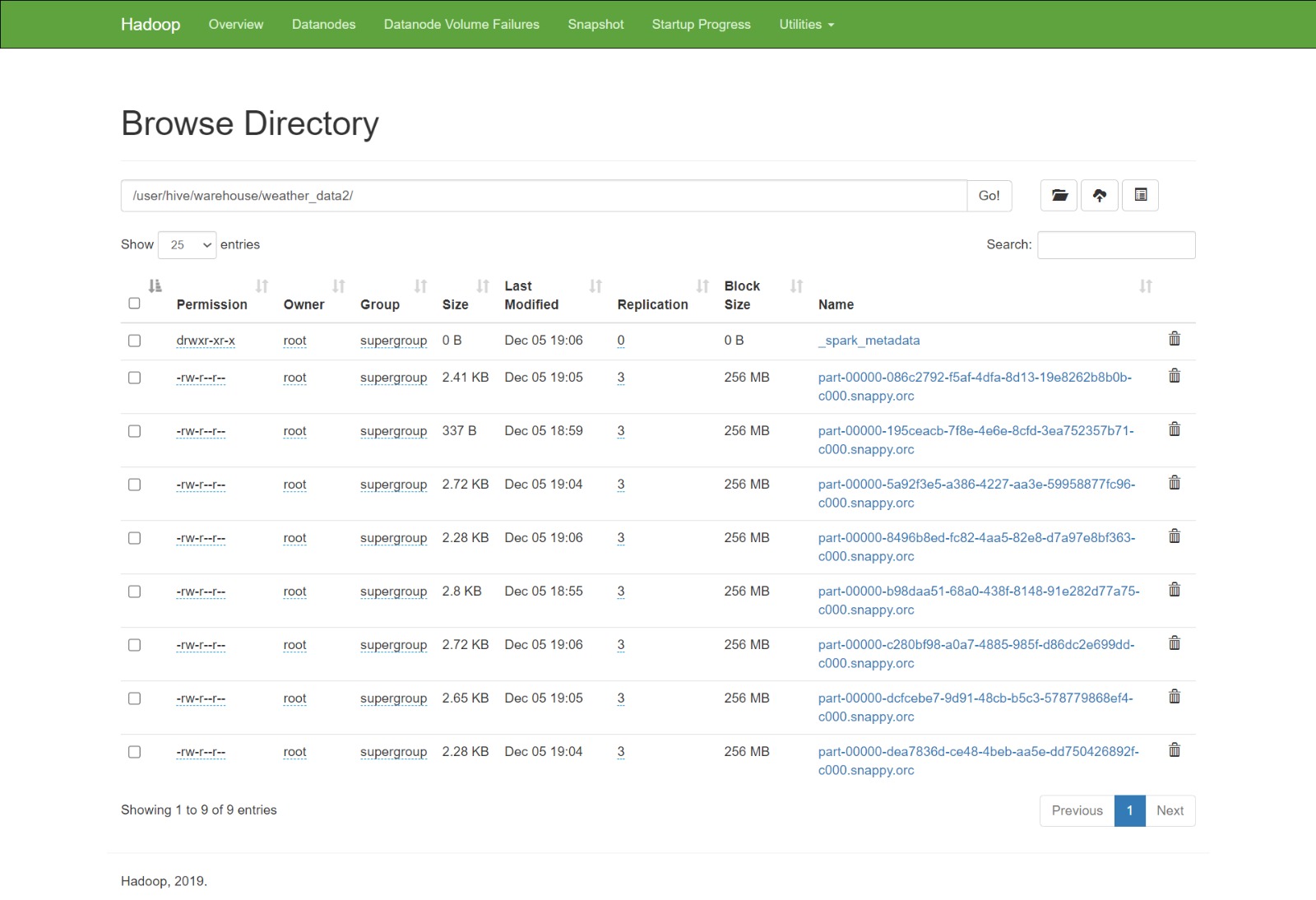


Figure 3.6 Vérification au niveau du namenode de l'ajout des données sur Hive

Par la suite vous pouvez vérifier dans la base de données pour s’assurer que les données ont bien été enregistré :

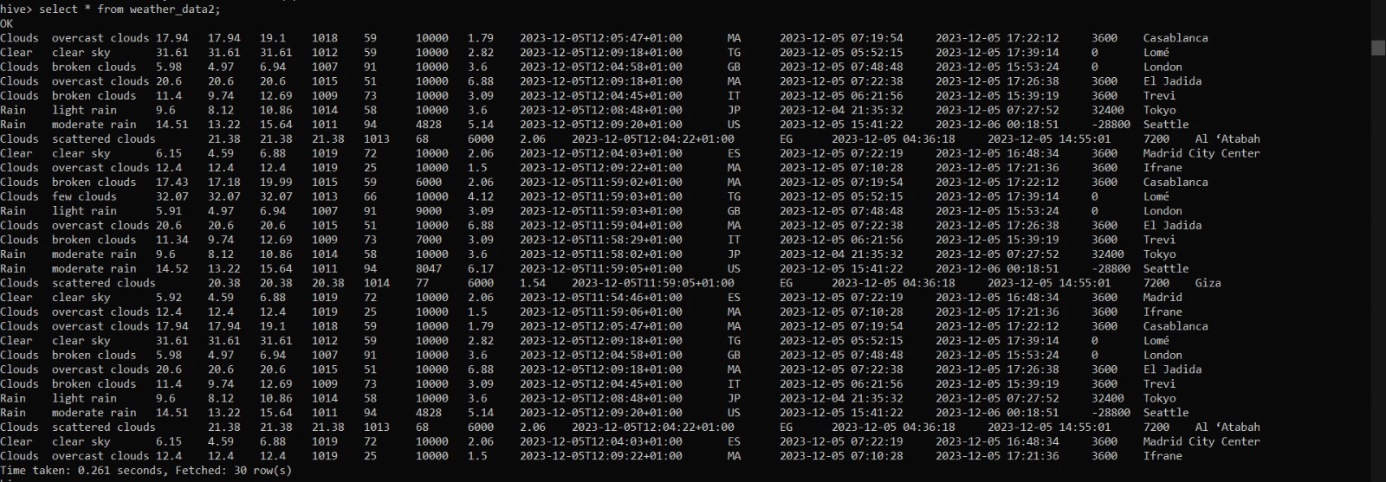


Figure 3.7 Vérification au niveau de Hive

## Visualisation des Résultats avec Tableau

La visualisation des résultats avec Tableau nécessite que les données météorologiques collectées soient stockées dans un format compatible avec Tableau, tel que dans une base de données Hive. Assurez-vous d'avoir suivi les étapes précédentes, notamment la configuration de Tableau pour se connecter à Hive.

Voici comment vous pouvez visualiser les données dans Tableau après avoir suivi les étapes du projet.

### Importation des Données

* Connecter vous à nouveau à Hive si nécessaire
* Une fois connecté à Hive, vous verrez les bases de données disponibles. Sélectionnez la base de données où vous avez stocké les données météorologiques.
* Choisissez la table ou la vue qui contient les données météorologiques.

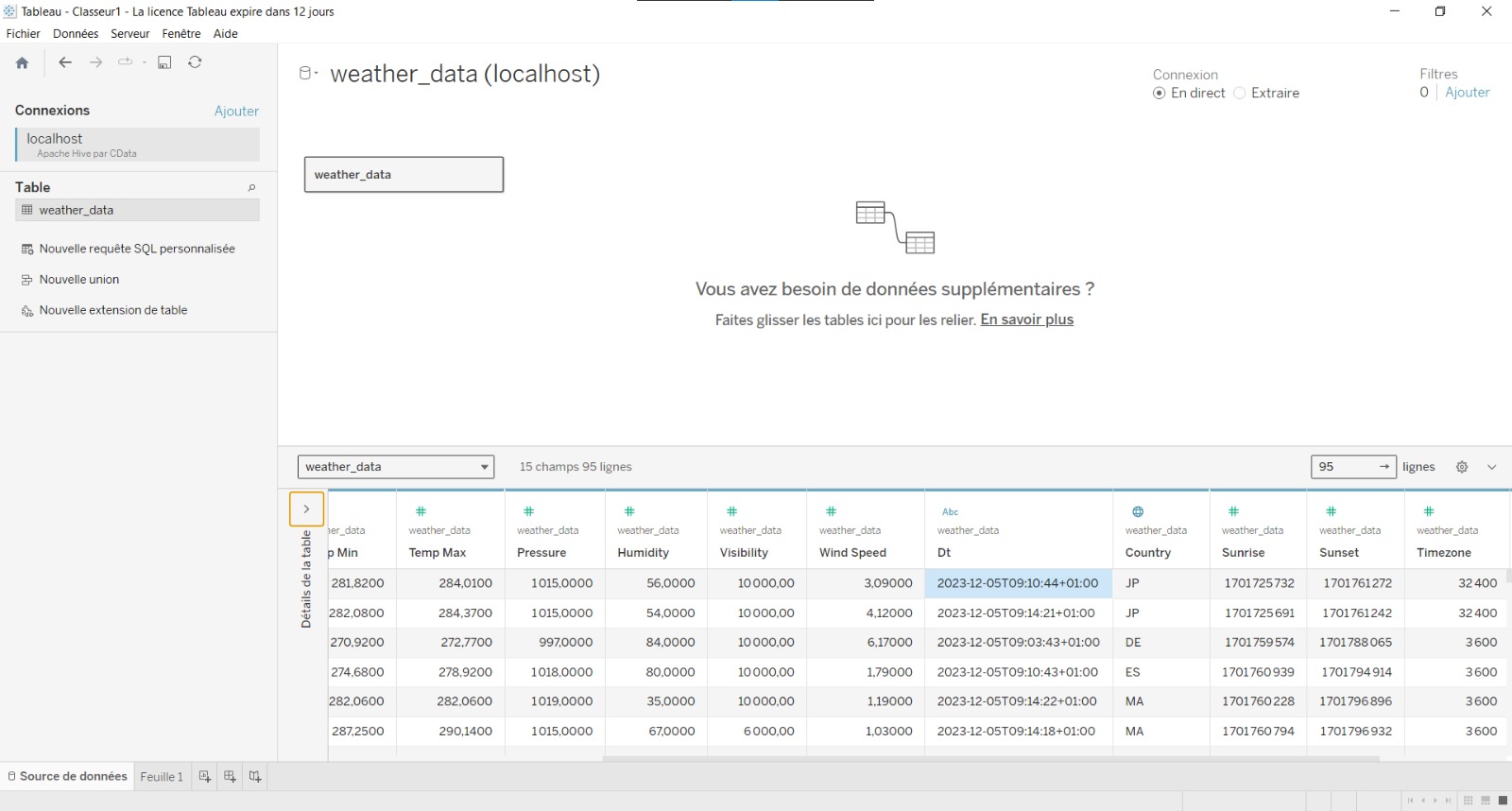


Figure 3.8 Importation des données de Hive sur tableau

### Création des Visualisations

* Allez dans l'onglet "Feuille" pour commencer à créer vos visualisations.
* Sur la gauche, vous verrez une liste de dimensions et de mesures (valeurs numériques). Faites glisser les dimensions et les mesures souhaitées vers les étagères "Colonnes" et "Lignes".
* En fonction des données que vous avez, choisissez le type de visualisation adapté. Par exemple, utilisez un diagramme linéaire pour suivre une tendance temporelle, ou une carte pour afficher des données géographiques.
* Ajoutez des filtres pour permettre aux utilisateurs de sélectionner des plages de dates, des villes ou d'autres critères spécifiques.

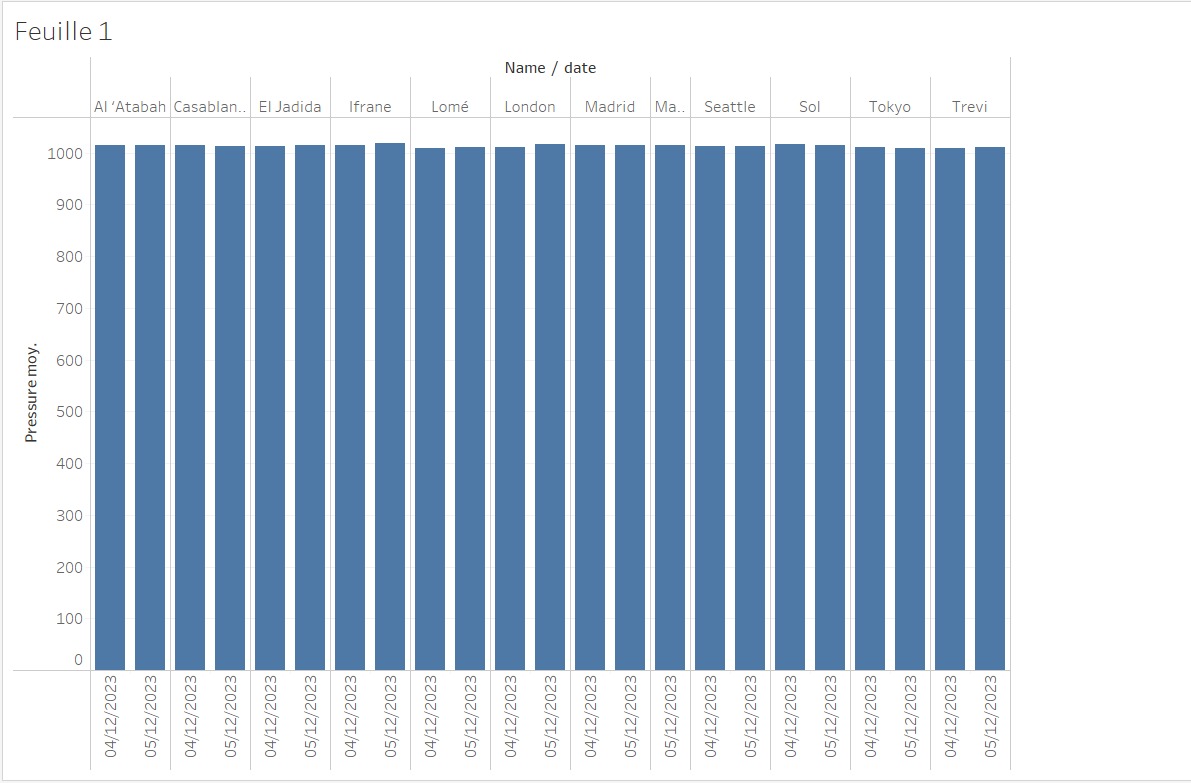


Figure 3.9 Visualisation de la pression par date pour chaque ville

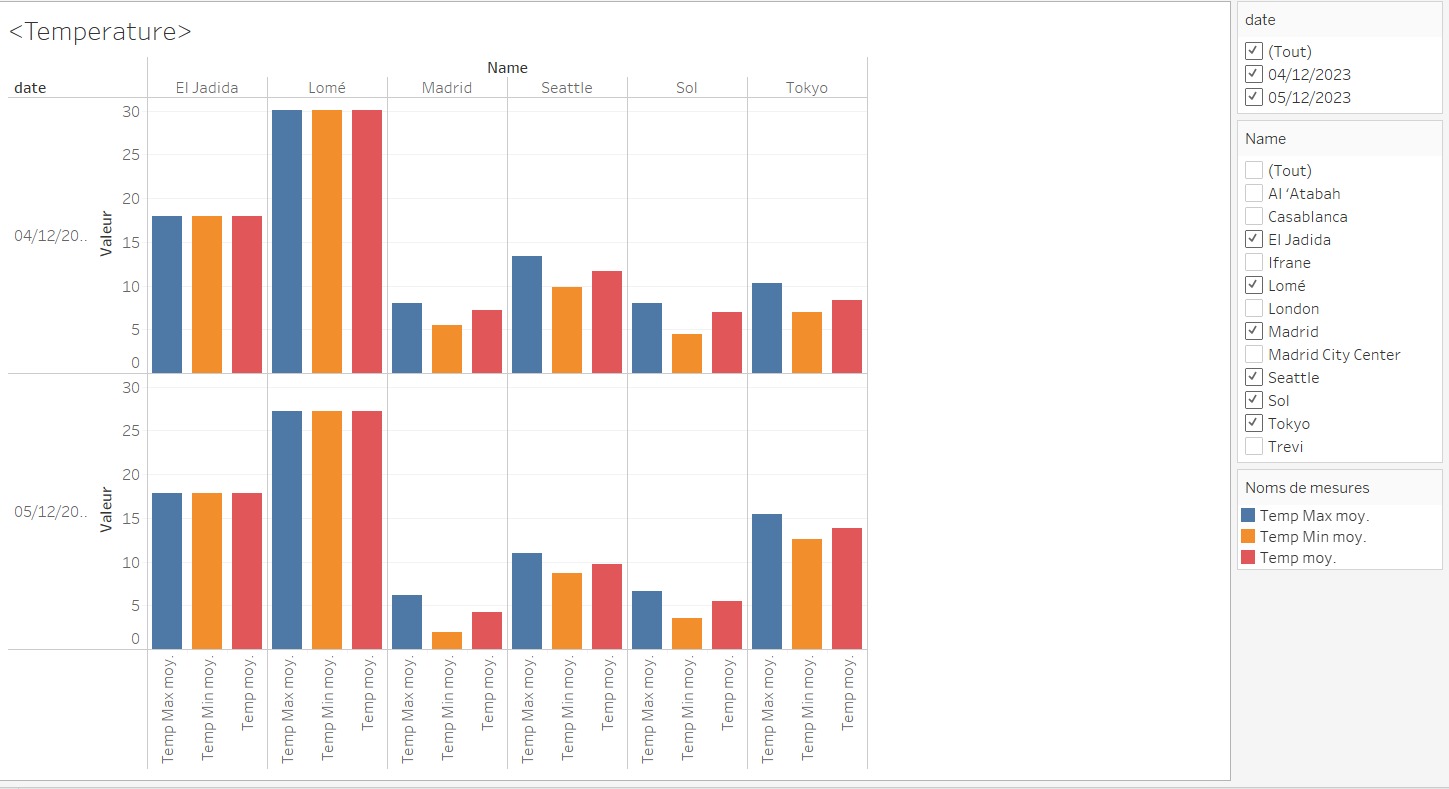


Figure 3.10 Visualisation des températures à chaque date pour chaque ville

### Création de Tableaux de Bord (Dashboard)

Allez dans l'onglet "Tableau de Bord" pour créer une vue d'ensemble qui combine plusieurs visualisations.

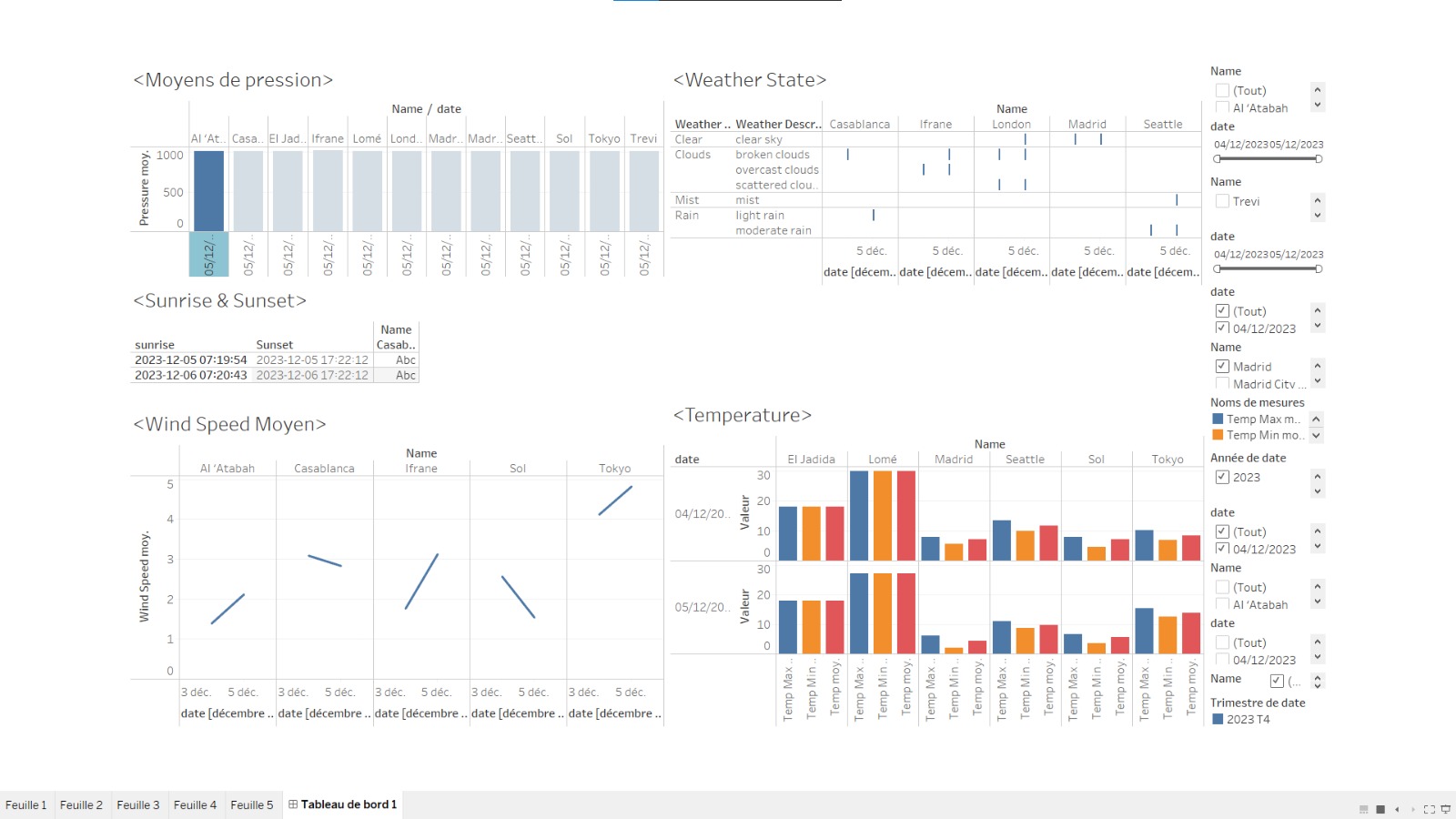


Figure 3.11 Tableau de bord représentant l’ensemble des visualisations

## Orchestration avec Apache Airflow

Airflow est une plateforme qui vous permet de créer et d'exécuter *des flux de travail*. Un flux de travail est représenté sous la forme d'un [DAG](https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dags.html) (un graphe acyclique dirigé) et contient des éléments de travail individuels appelés [Tâches](https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/tasks.html) , organisés avec des dépendances et des flux de données pris en compte.

Un DAG spécifie les dépendances entre les tâches et l'ordre dans lequel les exécuter et exécuter les nouvelles tentatives.

Les tâches elles-mêmes décrivent ce qu'il faut faire, qu'il s'agisse de récupérer des données, d'exécuter une analyse, de déclencher d'autres systèmes, ou plus encore.

### Interface utilisateur

Airflow est livré avec une interface utilisateur qui vous permet de voir ce que font les DAG et leurs tâches, de déclencher des exécutions de DAG, d'afficher les journaux et d'effectuer un débogage et une résolution limités des problèmes avec vos DAG.

Pour se connecter à l’interface :

* Nom d’utilisateur : airflow
* Mot de passe : airflow

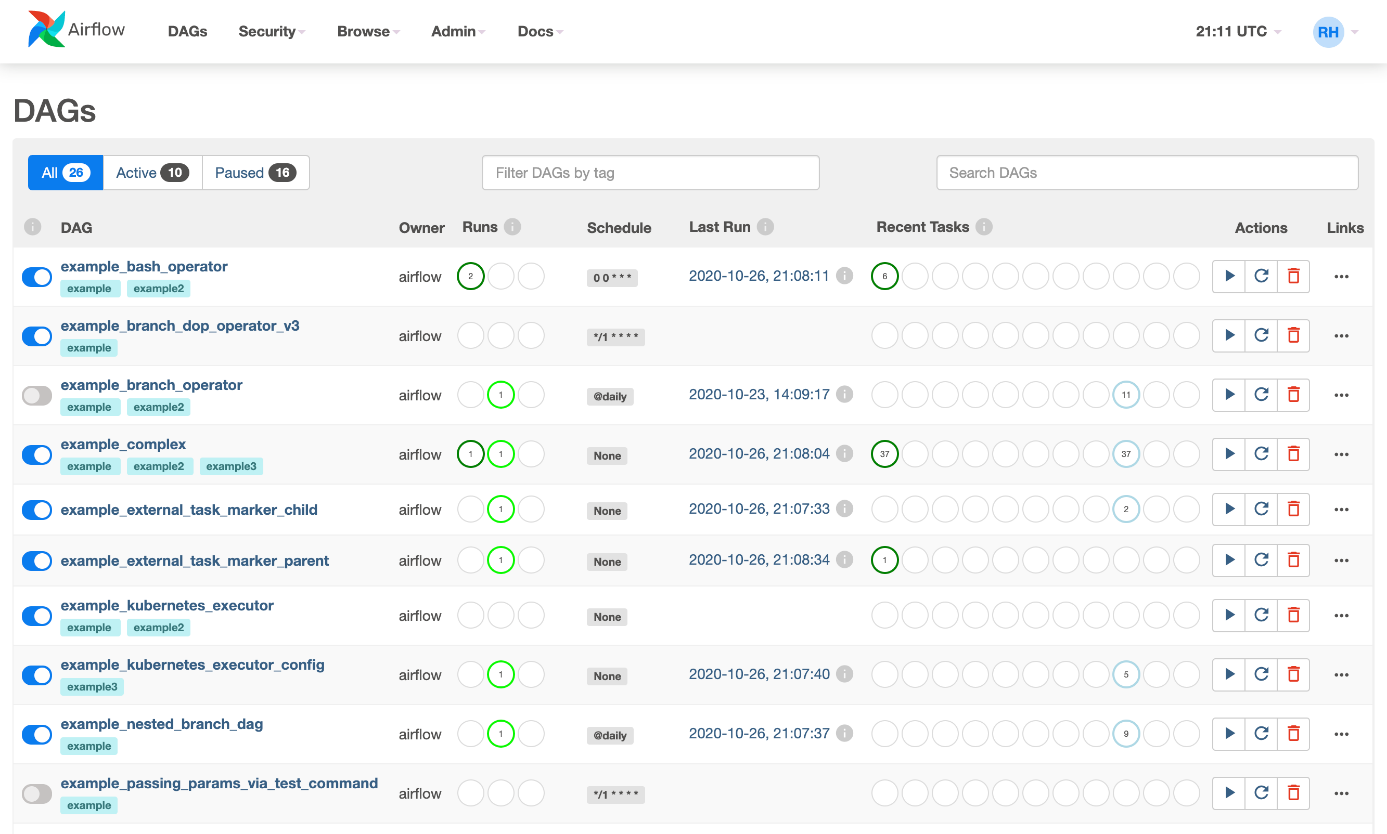


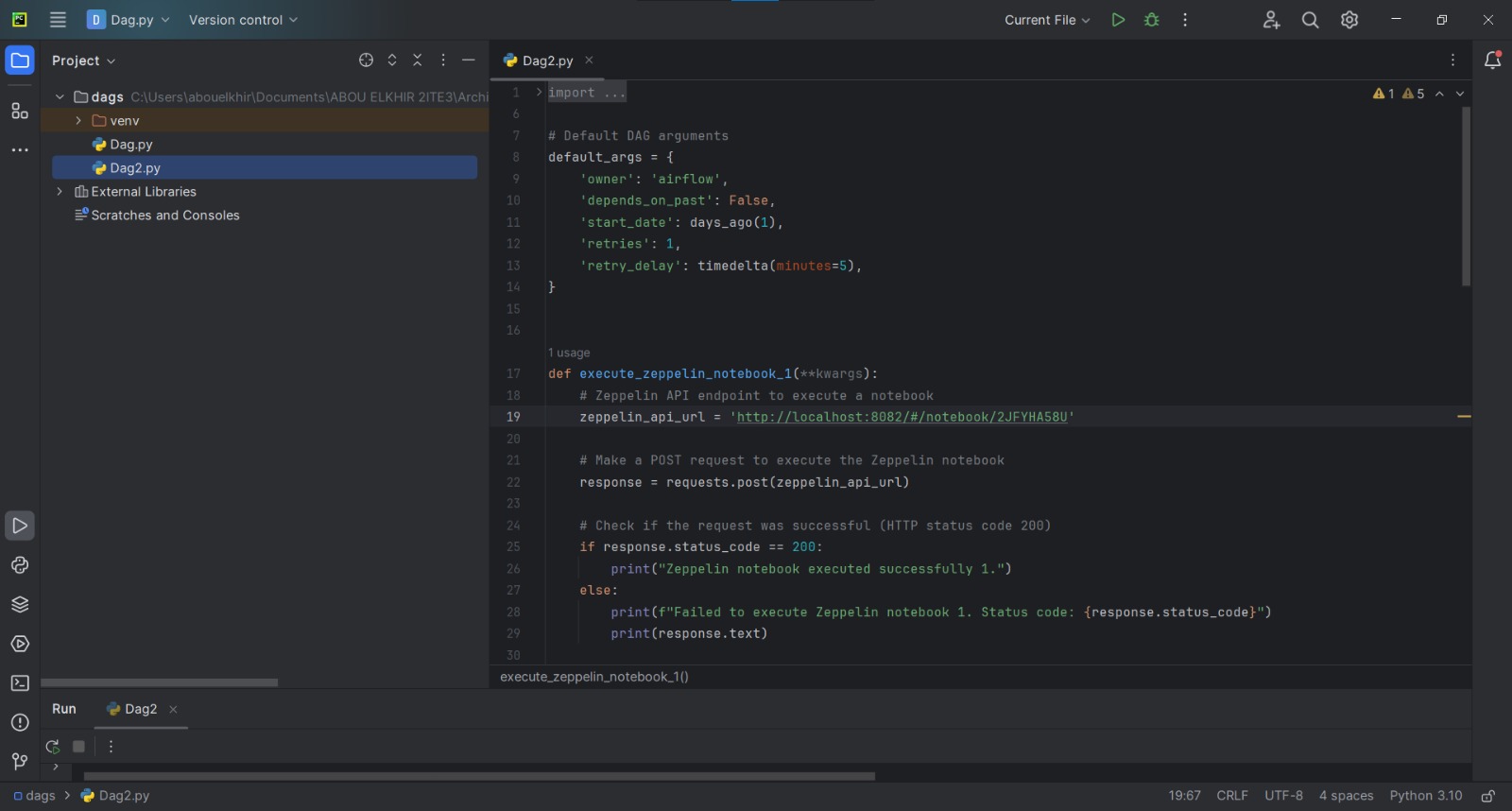
Figure 3.12 Interface utilisateur de Airflow

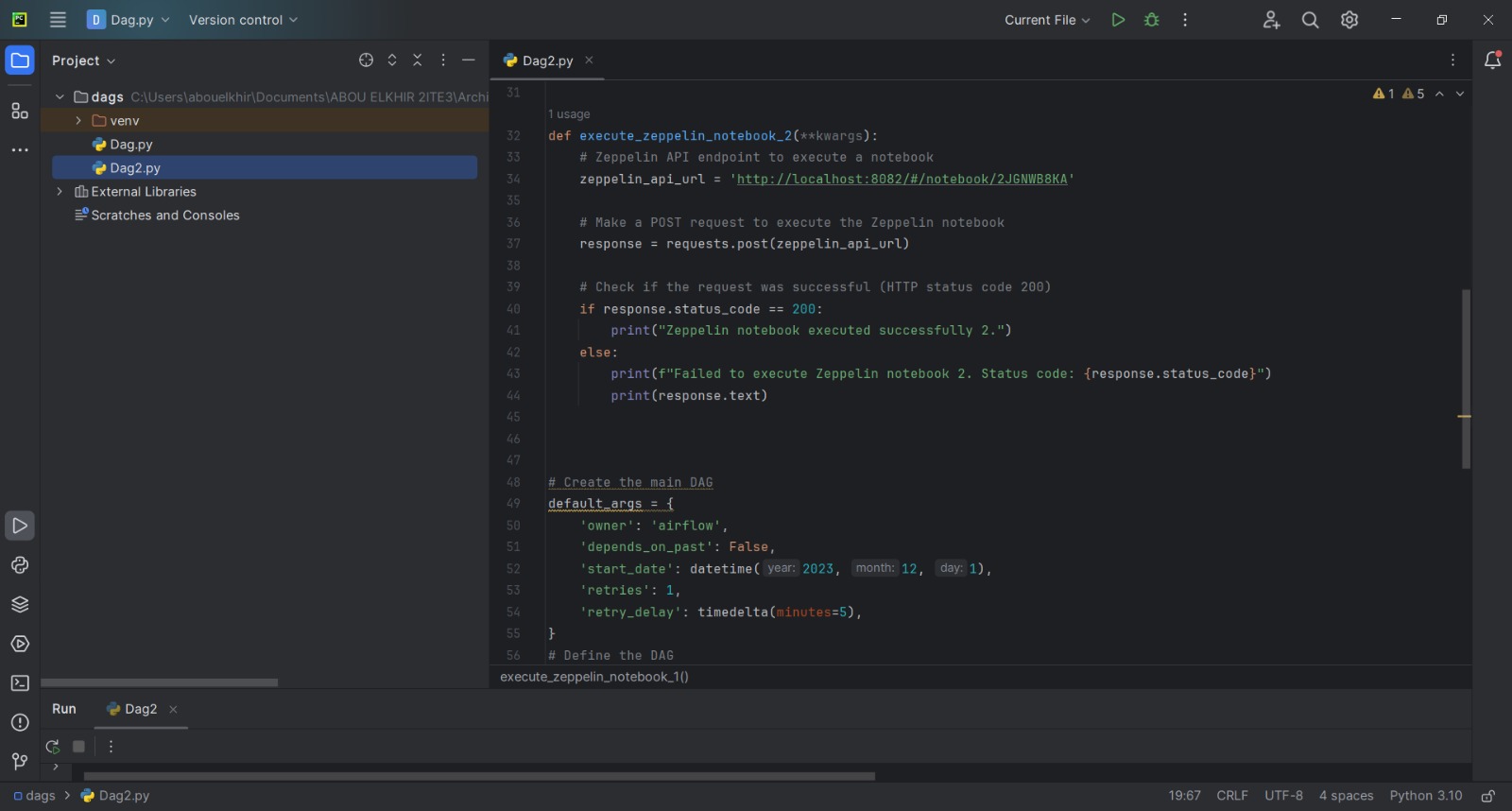
### Création et exécution du dag

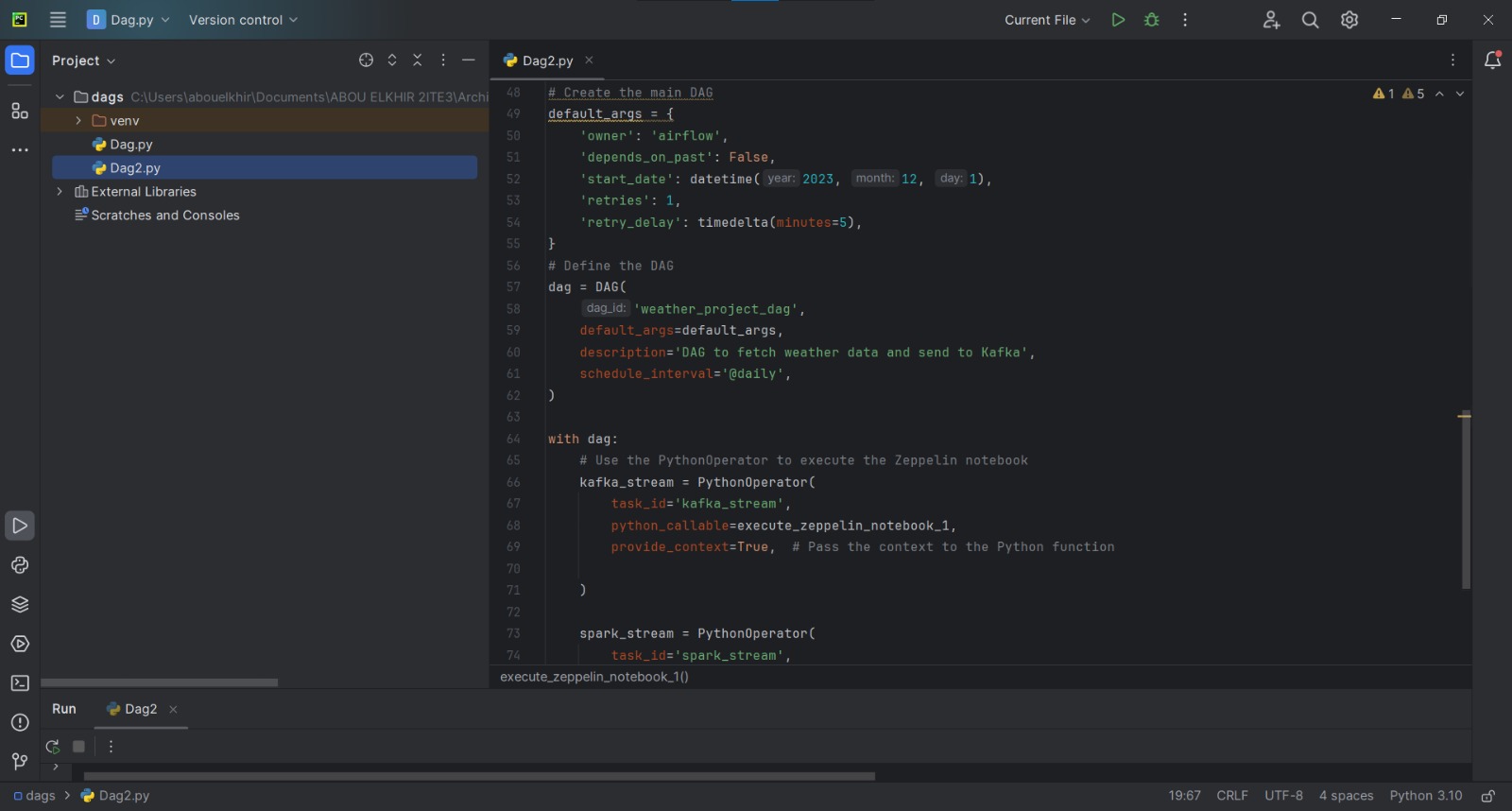
* Utilisez votre éditeur de code préféré (comme VSCode, PyCharm, etc.).
* Créez un nouveau dossier appelé "dags" à l'emplacement où vous souhaitez stocker vos DAGs.
* À l'intérieur du dossier "dags", créez un fichier Python pour votre DAG, par exemple, "dag.py".
* Dans ce fichier, écrivez le code décrivant les différentes étapes du processus.

Vous trouverez le code dans le fichier dag.py dans le dossier du projet, copier et coller le code dans votre fichier et faites les installations nécessaires.

* Dans le code du dag modifier les adresses des notebooks selon les id







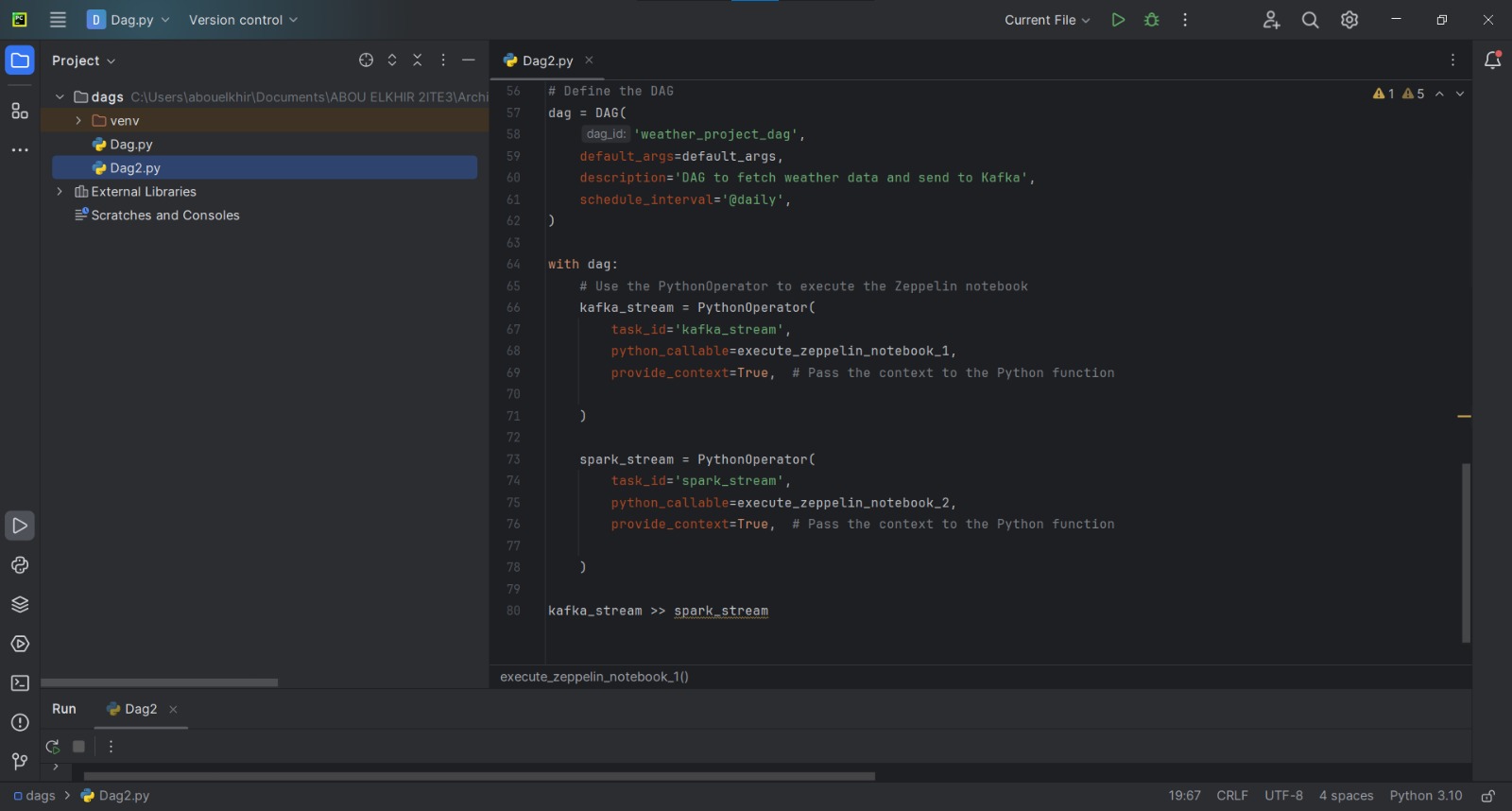


Figure 3.13 Code du dag du projet

* Assurez-vous qu'Airflow est en cours d'exécution et lancer l’exécution de votre dag.
* Airflow va automatiquement détecter et charger votre DAG.

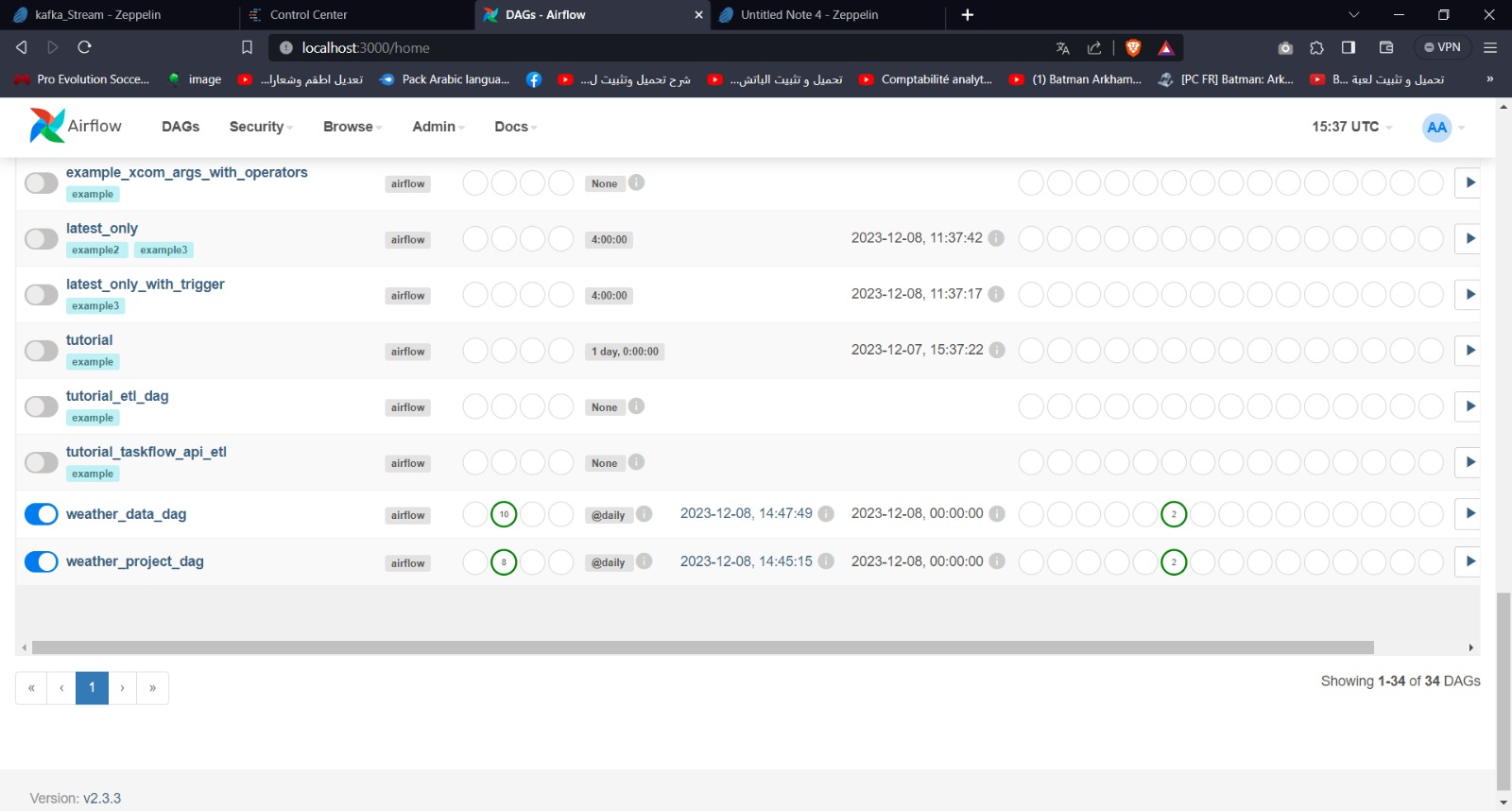


Figure 3.14 Vérification de l'ajout du nouveau dag à la liste des dags

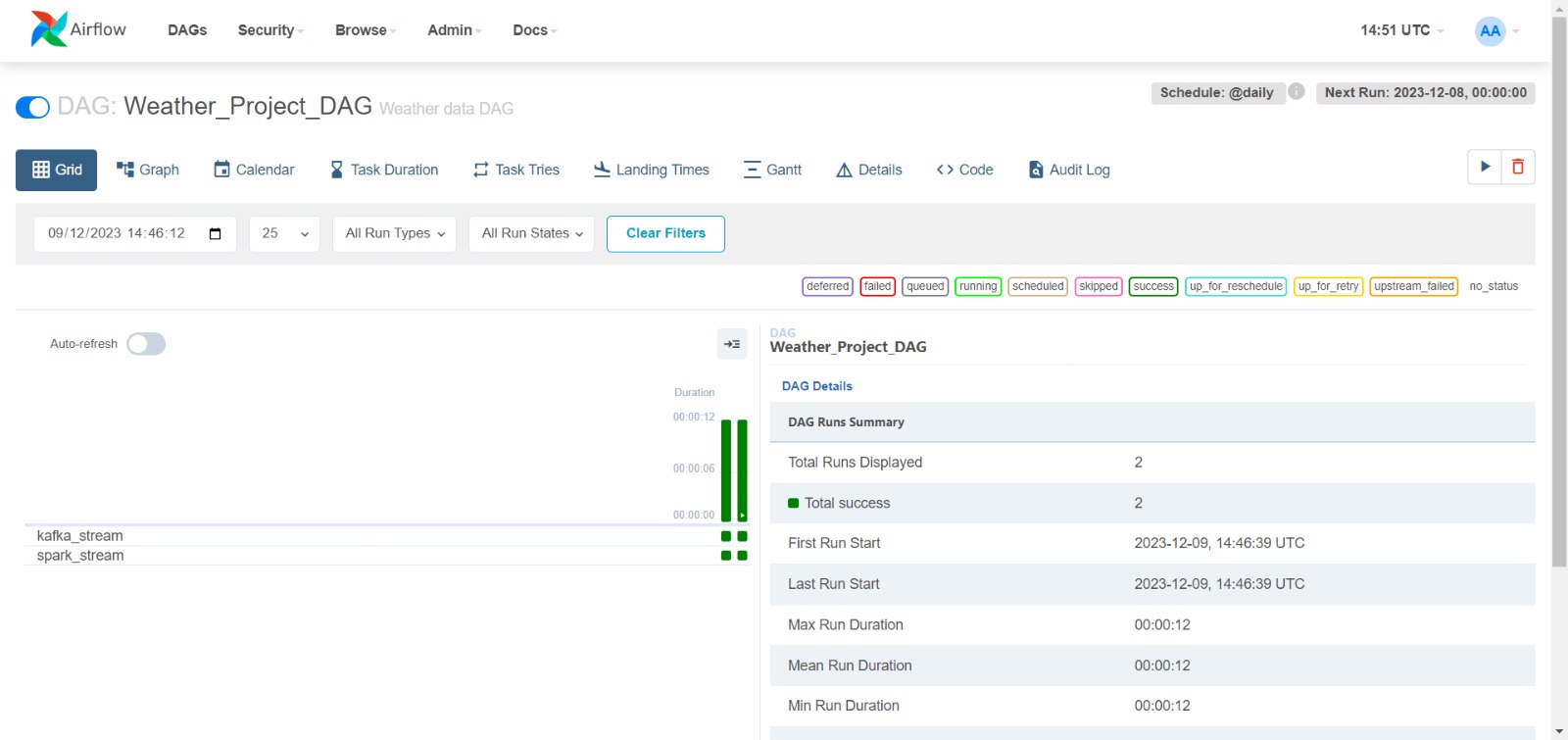


Figure 3.15 Visualisation de l'exécution du dag créer

Si vous avez planifié le processus de collecte, de traitement et de stockage des données à intervalles réguliers avec Apache Airflow, vous pouvez configurer Tableau pour actualiser automatiquement les données à partir de Hive.